

# A logical information system proposal for browsing terminological resources.

Annie Foret

IRISA, University of Rennes 1  
Campus de Beaulieu, 35042 Rennes cedex, France

foret@irisa.fr

## Abstract

This article presents an automated construction of a logical information context from a terminological resource, available in xml ; we apply this to the resource FranceTerme and to Camelis tool and we discuss how the resulting context can be used with such a tool dedicated to logical contexts.

The purpose of this development and the choices related to this experiment is two-fold : to facilitate the use of a rich linguistic resource available as open-data in xml ; to test and envision a systematic transformation of such xml resources to logical contexts. A logical view of a context allows to explore information in a flexible way, without writing explicit queries, it may also provide insights on the quality of the data. Such a context can be enriched by other information (of diverse natures), it can also be linked with other applications (according to arguments supplied by the context).

**Keywords :** Scientific terminology, Technological terminology, Multilingual applications, Information extraction, Textual data mining, Information retrieval, Linguistic resources, Open Data, Information Quality, Legal Information.

## 1 Introduction

This study aims to make linguistic data easier to exploit through the *logical information systems* approach : whereas such data are not always easy to use without assistance or expertise, logical information systems are especially designed to offer a flexible browsing of data when organized as a *logical context*. Some other works use a similar framework but their data are of different nature,

and their goals as well : (Cellier et al., 2011) apply Logical Concept Analysis to explore sets of patterns obtained by data-mining, (Quiniou et al., 2012) consider stylistic patterns, (Foret and Ferré, 2010) consider type-logical grammars, (Falk et al., 2014) uses several features including a thematic one to help identify new words.

In this proposal, we want both :

- to facilitate the use of a valuable linguistic resource (with a rich structure) and available in XML, and to allow its flexible querying and exploration without prior knowledge ;
- we want to test and consider a systematic transformation (*a transducer*) from such resources (in XML) to logical contexts ; such contexts can be loaded in a software allowing rich and flexible browsing on data, combining various heterogeneous criteria ; the way we represent the information in the context may also have an impact on its ease of use.

The aim is here to perform a transducer so as to present the data in a logical information system without losing information content, but gaining in ease of exploration. Other advantages are provided by a safe navigation (no dead-end property) and serenity.

The resulting context is freely available <sup>1</sup>.

**Terminological resource.** The selected resource concerns the scientific and technical fields, it also interests us for the richness of its structure : its multilingual aspects, with definitions, synonymous relations, etc. its confirmed status (with source and date of publication), variations according to domain/subdomains or according to linguistic criteria (several variants of English, for

---

1. at <http://www.irisa.fr/LIS/softwares>

example), possible absence or possible repetition of certain types of information.

This rich structure also allows further extensions : either with existing data or with new data that we organize in a similar pattern.

**Logical context.** A logical context is defined by a finite set of objects  $\mathcal{O}$ ,<sup>2</sup> and a finite set of logic descriptions  $d(o_i)$  expressed using a well-formed logical language  $L$ .

A *Logical context management system* can load and manage such a context, allowing querying a context by logical requests (explicit or interactive); then the answer is a sub-context of objects satisfying the query. We used *CAMELIS* (version 1)<sup>3</sup> for the experiment reported in this article. This software is based on *Logical concept analysis (LCA)* as defined in (Ferré and Ridoux, 2004). LCA is an extension of the formal concept analysis (FCA, see (Ganter and Wille, 1999)) : a *logical concept*, denoted  $c$  is a pair formed of an extent  $ext(c)$  (a set of objects) and an intent  $int(c)$  (a formula) such that the elements of  $ext(c)$  are exactly those which satisfy  $int(c)$ . These concepts form a lattice underlying the incremental *logical navigation tree* in the left window of the software. The software *CAMELIS* is also designed for managing sets of objects of different types. Object descriptions in a given logical context can have several origins : they can be retrieved by a transducer or come from extrinsic judgments (personal notes, for example); combining these modes allows to enrich the context and adapt it according to a user preferences.

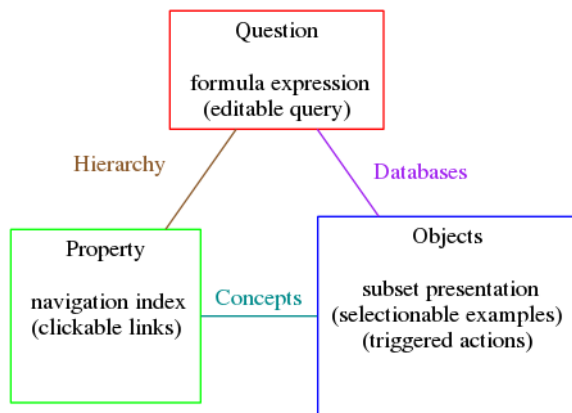


FIGURE 1: LIS and LCA

2. (several objects can have the same label)

3. <http://www.irisa.fr/LIS/ferre/camelis/>

Figure 1 illustrates how LCA generalizes Database and Hierarchical systems, the figure also follows the interface that enables three modes and shows synchronized related windows (as in figure 5) : a query on the top, links in the navigation index on the left, or objects on the right.

Thereafter, we present in section 2 our transducer implemented in XSLT<sup>4</sup> and we specify the construction methodology. We present in section 3 how to exploit the transducer output on the FranceTerme resource containing terms of different scientific and technical fields; we discuss several scenarios and benefits of this approach through this experiment. Additions and adjustments are proposed and discussed in 4 before concluding in section 5.

## 2 The transducer methodology

### 2.1 Some key aspects

The transducer is designed to present data in a logical information system without losing information content, but gaining in ease of exploration. The diagram in figure 2 illustrates the approach, where the automated steps (solid arrows) are distinguished from manual or semi-manual ones (dotted line).

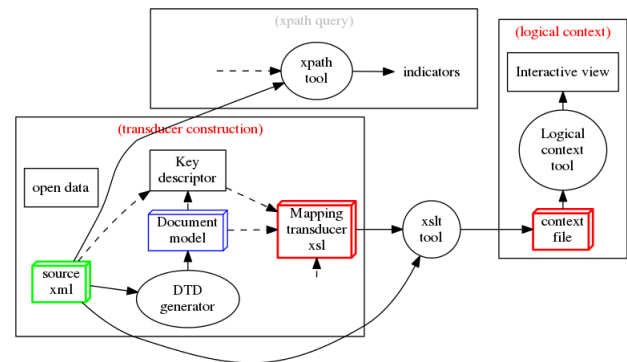


FIGURE 2: global architecture

**Source Document Schemas.** The transducer is designed to apply automatically on documents conforming to some document specifications. In general, such a specification can be automatically produced from an XML instance. We used DTD

4. a web availability is planned.

**Logical information system capabilities.** We recall that a logical context system must allow the loading and management of a context described by its objects and its objects properties. More precisely, we assume that :

- the logical context allows for some *inferences* : at least from classical logic (such as *if A then A or B*) possibly with axioms useful to model the context and organize its presentation (for example to reflect a taxonomy and only see some salient properties) ;
- some general form of information retrieval and multi-faceted means are provided, as "logical facets" and "logical criteria" combinations ;
- a *modular and dynamic* construction is allowed, for both sets of objects and sets of properties.

**CAMELIS system.** In our experiment, we used the logical context tool **CAMELIS**, that, to our knowledge, is the only logical context management system. The tool interface displays three connected windows (see figure 3) :

- a query window (top) ;
- an object window (right) ;
- an index window (left) as a *navigation tree*.

Properties in the navigation tree are organised as a clickable summary grouping hierarchy properties : it is important to note that a navigation link there corresponds to a *sub-context* (as in figure 5, the link/sub-context cardinality is given and a color is also associated with a concept :two navigation links with the same color lead to the same sub-context).

In the browsing mode, this tool allows three forms of query that select a *sub-context* :

**expert/query mode** by *editing in the query window* ; the displayed objects become those satisfying the logic query ; the navigation tree is then presented in a form adapted to the new context ;

**example/object mode** by *selecting a set of objects in the object window* ; the query automatically becomes an expression for the common properties of objects ;

**index/property mode** by selecting a property (or more) in the navigation tree ; the objects

displayed become all those verifying the selected property (links) ; the query window is then automatically updated.

**CAMELIS general properties.** The *consistency* between the three windows is ensured. In addition, a session will not lead to an empty set of objects when following the links in the navigation tree : this important property is the *navigational safety*.

**CAMELIS update and logic modularity.** Using the same interface, we can add and dynamically update objects and properties, then export as a new *logical context file* (useful for example to generate a documentation for the objects of a selected sub-context). The tool can also be adapted to choose a dedicated logic, obtained by combination of logic functors (Ferré and Ridoux, 2004).

We do not detail these last aspects in this article.

**Control.** Part of the context information visible in the tool, can be retrieved by other means. We built some XPath queries to control the process and to produce complementary indicators.

We also built a control-context (figure 7).

## 2.2 Transduction overview

We give here the characteristics and main stages of realization of the transducer, in its basic version. This construction is guided by the information in the DTD generated by XPath queries and control.

**Key selection.** Defining a key in the source (by means of its document schema or of an XPath expression) is a preliminary stage, and the key definition plays a central role in the context construction. For this experiment with *FranceTerme.xml*, we considered `//Article/@id` (XPath).

**Components.** The program is designed to facilitate its updating, structured by source components and similar typical treatments. The treatment of a given source component depends on its kind (XML element, attribute) the relevant part of context, its status (optional, repeatable or not), the domain of the source content, and the desired rendering (data type, property name, property hierarchies).

**Main Loop.** For each source item *Article* :

- each object get a unique label (extracted by `Terme[@statut='privilegie']`), used for the object presentation and for a string property in the navigation index ;

5. it is available at <http://saxon.sourceforge.net/dtdgen.html> ; the terminological XML source file we used is accompanied with an XML Schema xsd, but without guarantee

- the key becomes a number property  
`articleID = ... (xslt6);`
- the publication date is processed to appear in the index at different levels of date detail ;
- most other components are processed to produce strings of the form :  
`property_name is "string_value"`
- property names may depend on several XML components (such as `Terme` element with a `@statut` attribute), they are organized to allow their grouping and multiple levels of detail (`Terme ?` is more general than `Terme SYNONYME`);
- we also give a common prefix `_Plus` to properties for data in the source, but not visible in the ressource site (such as data about committees);<sup>7</sup>
- for XML elements that can be repeated for a given key/object, (such as `Terme`) we use an inner loop ;  
*at this stage the output file context contains the description of one object per line, with its main properties ;*
- other components (such as foreign equivalents or antonyms, optional or repeated) are rendered by rules of the form :  
`rule_extr (key=id) --> (prop1 is val1)`  
that automatically associate the property to the object designated by the key.

## 2.3 Modularity of context

For treating a logical property related to an XML component, such as `<Attention>` child (optional and repeatable) of an `<Article>` identified by its attribute `id`, several alternatives are possible :

-to indicate the name of property and its value by assembling and repeating the property name for the object, using this pattern :

```
mk "object" key=id, ..., is prop1
val1, val2 prop1 is ... is ... prop2
```

-to indicate each property value by transformation rules, using this pattern, when the object is assumed to be already created and associated with the key :

```
rule_extr (key=id) --> (prop1 is val1)
```

6. `<xsl:value-of select="concat('articleID=' $varArtId)"/>`

7. in the navigation tree, compound names of the properties are grouped by prefixes, details appear by opening a link `_Plus ?`

```
rule_extr (key=id) --> (prop1 is val2)
```

this will automatically add each property value to the object with key `id`.

This second solution brings some modularity since we can put rules in separate files, the properties being effectively added to objects after a file import. We chose this approach by means of rules and keys for some repeatable components.<sup>8</sup>

## 3 Logical Context and facets

In this part, we discuss several possible search modes in the resulting context, where navigation links (incremental) correspond to logical concepts that can be selected.

### 3.1 Simple searches on several data types.

**Multilingual data.** The `FranceTerme` resource contains translations in several languages, with variations for the same language. Those data are attached to various domains and subdomains (possibly several ones for a given object).

*Scenario.* An exploration of the logical context can be conducted that way, for example :

- open the `Domaine ?` property in the index ;
- select-click `Domaine is "Informatique"` (computer science), this yields the corresponding sub-context (with 3 coherent views) ;
- we may further select-click `Domaine is "Droit" (Law)`, also automatically expressed as `Domaine is "Informatique"` and `Domaine is "Droit"` in the top window ;
- open the `Equivalent ?` property then open `Equivalent _en is "..."` in the index etc.
- open the `PubliArticle ?` property then `PubliArticle date = 2014` in the index etc.

Another simple search on "streaming" is shown in figure 3.

**Sub-context Cardinalities.** In the property index-tree, we may choose an order for displaying a given facet. This is useful for example to read directly which `Equivalent _en` correspond to the greatest number of French terms (figure 3).

8. Here is a typical xslt fragment (with some special symbols treatments for compatibility) :

```
<xsl:for-each select="Attention"> <xsl:value-of
select="concat($varRulePart1,'idArticle=',
$varArtId,$varRulePart2, 'Attention is ', $varQuote,
translate(normalize-space(./text()), $varQuote, $varBQuote),
$varQuote)"/> <xsl:text> </xsl:text> </xsl:for-each>
```

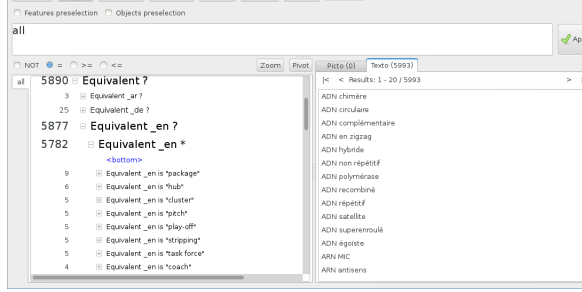


FIGURE 3: open facet en, before selection

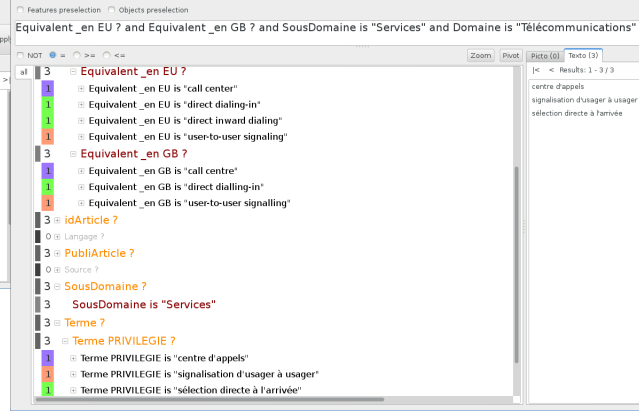


FIGURE 5: Variants

**Data types.** Data types other than attributes and strings can be handled, Figure 4) shows a possible use of dates, allowing for more or less fine-grained selections.

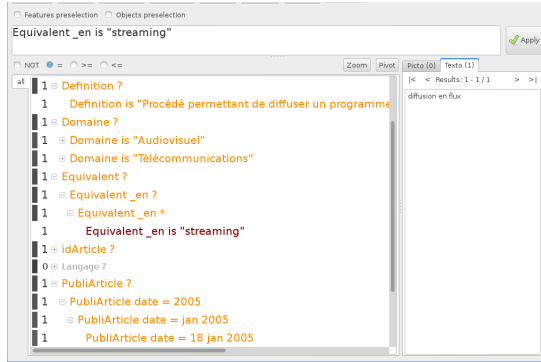


FIGURE 4: Facet en selected, facets date and domain opened

kind of subcontext summary and extra informations (here *Attention* is the focus element).

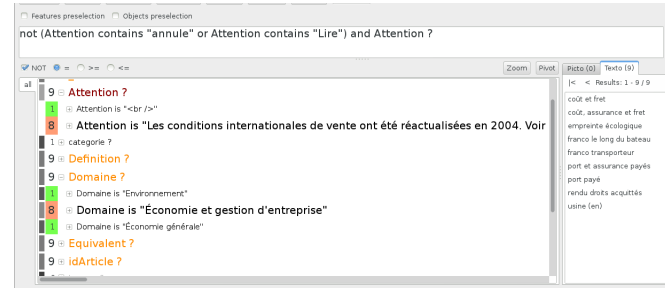


FIGURE 6: Focus on *Attention*

**Exploring variants and false-friends.** Figure 5 comes from selections in the index tree ; links of a same color characterize the same set of objects (concept).

This example illustrates the identification of potential linguistic errors (in a domain / subdomain)

Other examples such as "package" (*Equivalent\_en*) or some abbreviations ("ABS") can be highlighted as ambiguous : the dynamic navigation links (domains, etc.) then provide hints to disambiguate.

### 3.2 Focused search on elements and exceptions : summaries.

Using the *Not* button at a given stage, we can arrive at a subcontext characterized by  $A_2 = \text{not}(A_1)$  and  $A_0$  as in Figure 6. Property  $A_2$  expresses a search for exceptions to  $A_1$ , we get a

### 3.3 Other scenarios : data quality

This navigation mode allows to detect abnormalities, in particular pseudo-empty properties appear on other facets (through a link like *Definition* is "") these cases often correspond to existing but empty XML elements (but are not XML errors). Low cardinalities in the navigation tree may suggest to explore the link, by selecting it and opening other facets simultaneously ; we can analyse this way "the words without translation, following the *not Equivalent?* link.

In case of redundancies, these may become easily noticeable through browsing : exploring the *Antonymes* facet, we can see XML structuring redundancies (this information being carried by two source elements).

### 3.4 Control and actions from a context

The logical context software can assign actions to objects by properties ; clicking on an object label then shows a contextual action menu.

This is useful in particular to inspect objects in their source xml file.

Other kinds of control (for coverage, counts, etc.) are made

- by XPath queries on the XML document used to verify if certain characteristics of particular sub-contexts (planned or explored) are consistent with the source document ;
- a meta context built, following the DTD schema, whose objects are : element names, and the pairs (attribute name, element name). These objects are associated with actions parameterized by their label, in our case (Figure 7), the action is an XPath query using BaseX ; This can be adapted easily to another set of controls.

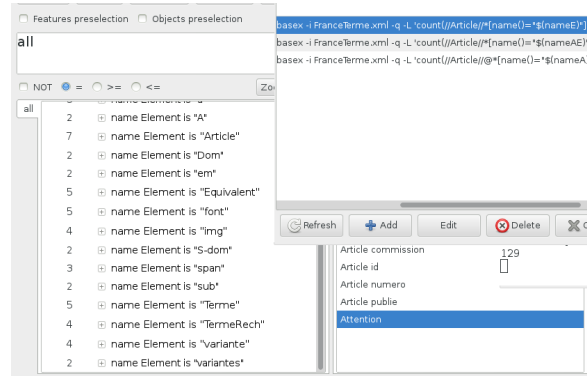


FIGURE 7: control (meta) context

## 4 Refinements and user preferences

A logical context tool such as CAMELIS by its genericity and its features, allows many alternatives to represent and use a terminological resource like FranceTerme.

Some initial choices can be easily revised or completed ; for example the name of a property can be changed directly through the interface (or by simple transformation of the context file).

The choices and refinements should provide a better context. Several quality criteria can be considered : effectively obtaining a desired result (usefulness / completeness) ; the number of steps to get there (effectiveness) ; rich browsing indexes (multiple views) and efficient indexes to pursue

the navigation (through the proposed increments) ; a flexible mode of interrogation and ease of interpretation.

We indicate some possibilities through modifications in the experiment.

### 4.1 Adapting facets using rules

Domains and SubDomains data have been translated. The resulting two context files contain update rules of the form :

```
rule_extr Domaine is "Acoustique" -->
```

```
Domain is "Acoustic"
```

when loaded in the context, properties on the right hand side are added to all objects verifying the left hand side.

### 4.2 Improving grammatical categories using rules and axioms

In the original context, we can see (with an appropriate ordering) that among the terms with a category attribute, the names (categorie is "nm" or categorie is "n") are the majority, followed by categorie is "adj.". However these grammatical categories are listed with various values, we can observe : which may include in particular :

- a disjunction, as in : categorie is "adj. ou n.m." and Equivalent\_en is "crossmedia (n. ou adj.)" which selects the term transmédia ; but we also observe its permutation categorie is "n.m. ou adj." ;
- a more or less fine granularity, as in : categorie is "n.m.inv."

The addition of rules and axioms in the logical context permits to harmonize these properties, resulting a more structured navigation tree according to this facet. A few lines in the resulting context define a hierarchy of categories, such as :

```
rule_extr categorie is "n.m.inv." -->
Categorie_n_m_inv
...
Categorie_n_m_inv axiom, Categorie_n_m
Categorie_n_m axiom, Categorie_n
...
```

Note that such improvements could apply to other terminological resources and result from linguistic analysis or other lexicons.

### 4.3 Axioms for property variants

We have seen that a property Terme? covers three statutes (PRIVILEGIE, SYNONYME, ANTONYME).

A context of axioms can facilitate a search on all or a subset of these variants :

```
axiom PRIVILEGIE, SET
axiom SYNONYME, SET
axiom SET, ANY
axiom ANTONYME, ANY
```

A query may then group several properties (having a status below another expression like SET) as exemplified in figure 8.

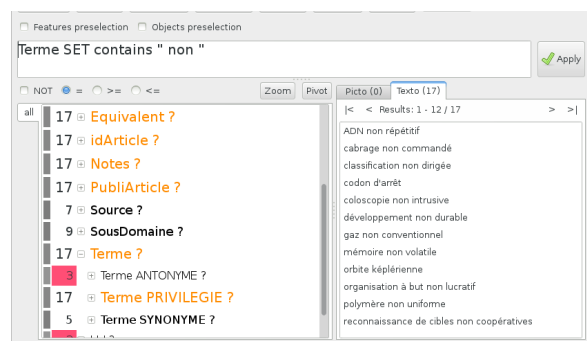


FIGURE 8: with axioms on property names

This example shows a query for Terms (including status variants) containing negation ("non").

Note that such axioms can be added or modified in a modular way.

#### 4.4 Linking data and resources

##### At the property level in the navigation tree.

The website *FranceTerme* allows to select new terms, from the description of a current term (denoted as  $t_1$ ) by a link *See also*. This is rendered in the context navigation tree by the facet *Voir aussi* (See also) is "... information on  $t_j$ " (denoted as  $f_j$ ) where the term to see  $t_j$  is shown with its key *Article*. Two modes of translation of this piece of information have been tested :

- in a basic mode, as a simple property  $f_j$  of the current term  $t_1$
- in a full (reflexive) mode, where  $t_1$  has  $f_j$  and also  $f_1$  : *Voir Aussi* is "... information on  $t_1$ ".<sup>9</sup>

This second mode allows this type of scenario : while  $t_1$  has property  $f_j$ , select this  $f_j$  link in the navigation tree ; by reflexive closure,  $t_j$  is also in the sub-context, and can be further selected.

We treated in the same way the reciprocal link of *Voir Aussi*, by adding (by the transducer) a *Voir Depuis* (See From) "... property. This enables to group into a sub-context the terms poin-

ting to (or pointed to by) a particular word (or more).

##### Notes.

- In the context for *FranceTerme*, we observe some terms (15) satisfying this query :

*Voir Aussi?* and not *Voir Depuis?*

these are the "terms pointing to an article, but not pointed to" ;

- according to the resource schema, other elements (synonyms, antonyms, ...) could be treated similarly, with navigation links in context.<sup>10</sup>

**Linking to other resources at the level of actions.** As explained about control, the logical context management software allows to associate actions to objects.

We can use this mode to associate an object with a process (or more) on this item that may be introduced in the interface from a context menu related to the object (or group of objects). The setting can be provided at the transducer level. A file describing these actions can be later loaded from the interface.

We generated connections to :

- a parser, installed locally : the processing chain (open) *Bonsai* (Candito et al., 2010) which takes as parameter the label of the object ; a selection of this action on the object provides a syntactic analysis of the label expression ;
- a web link to another terminological resource for French (CNRTL, <http://www.cnrtl.fr/>) the parameter being the term as above ; a selection of this action on the object opens the browser on the website page for this term, if there exists one (none for some *FranceTerme* expressions) ;
- an XML link to a subpart of the source file, through an XPath tool (BaseX) the parameter being the object key (attribute *id* of *Article*) ; a selection of this action on the object executes the software with a prepared BaseX request using the object key.

This action list is not exhaustive and can be adapted. In particular, we could consider links (local or not) with other analyzers, or other linguistic resources and retrieve results to enrich the logical context with new properties. The capabilities of *Full text search* (of BaseX) could also be ex-

9. no addition to the terms that have no link

10. this treatment is not currently done for the other elements (in the source xml these terms do not necessarily correspond to an article/object).



ploited.

## 5 Conclusion

The general aim of this proposal was to show how a logical concept analysis (LCA) framework and tools could be beneficial for browsing terminological resources ; through this experiment the purpose was twofold :

- to facilitate the use of a useful language resource (rich structure) and available in XML,
- to envision a systematic transformation of such resources as XML to logical contexts.

Improvements may also be suggested and brought to the data ; other treatments may also be eased, for example a selected sub-context can be exported as text and generate other results (such as a documentation).

We illustrated how a logical context allows to explore linguistic information, in a flexible way, without a priori knowledge, and also get guidance on data quality (in the navigation tree, counts and colors for concepts, ...) New linguistic information (personal, enterprise, ...) could be incorporated easily in the initial context (if they comply with the document model and the key assumption).

Additional data to compare and enrich the content can also be added in several ways and for many languages, (for French : Wordnet Wolf (Sagot and Fiser, 2012), Lefff lexicon (Sagot, 2010), etc.) :

- by adding objects without confusion between sources (since a property indicating the source is associated with the object) ;
- by adding properties to expand the browsing possibilities ;
- by adding triggered actions on objects.

Other actions corresponding to linguistic processing can be added to the context : parsing the expression (several languages), syntactic head, etc. We could also consider inverse properties (such as translation) and enrich the context with these objects.

Moreover, it seems that the development method could be transposed to other open data and linguistic xml ressources. To some extent, the construction of the transducer presented here could be automated if it relies on a determination of a key and a grid indicating for a *source component*, its label, its type, its repeatability, and the way it should be rendered. A similar experiment

could be carried out by adapting the standards and software tools of the semantic web. Finally, we mainly discussed browsing, future work could also concern updates.

## References

- Marie Candito, Joakim Nivre, Pascal Denis, and Enrique Henestroza Anguiano. 2010. Benchmarking of statistical dependency parsers for french. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 108–116. Chinese Information Processing Society of China.
- Peggy Cellier, Sébastien Ferré, Mireille Ducassé, and Thierry Charnois. 2011. Partial orders and logical concept analysis to explore patterns extracted by data mining. In Simon Andrews, Simon Polovina, Richard Hill, and Babak Akhgar, editors, *Conceptual Structures for Discovering Knowledge - 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK, July 25-29, 2011. Proceedings*, volume 6828 of *Lecture Notes in Computer Science*, pages 77–90. Springer.
- Ingrid Falk, Delphine Bernhard, and Christophe Gérard. 2014. From non word to new word : Automatically identifying neologisms in french newspapers. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odiijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 4337–4344. European Language Resources Association (ELRA).
- Sébastien Ferré and Olivier Ridoux. 2004. Introduction to logical information systems. *Inf. Process. Manage.*, 40(3) :383–419.
- Annie Foret and Sébastien Ferré. 2010. On categorical grammars as logical information systems. In Léonard Kwuida and Baris Sertkaya, editors, *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*, volume 5986 of *Lecture Notes in Computer Science*, pages 225–240. Springer.
- Bernhard Ganter and Rudolf Wille. 1999. *Formal concept analysis - mathematical foundations*. Springer.
- Solen Quiniou, Peggy Cellier, Thierry Charnois, and Dominique Legallois. 2012. What about sequential data mining techniques to identify linguistic patterns for stylistics ? In Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing - 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Pro-*



- ceedings, Part I*, volume 7181 of *Lecture Notes in Computer Science*, pages 166–177. Springer.
- Benoît Sagot and Darja Fiser. 2012. Cleaning noisy wordnets. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012*, pages 3468–3472. European Language Resources Association (ELRA).
- Benoît Sagot. 2010. The lefff, a freely available and large-coverage morphological and syntactic lexicon for french. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.